



Тернопільський національний технічний
університет імені Івана Пулюя

Кафедра автоматизації
технологічних процесів
і виробництв

Лабораторна робота № 8
з курсу
”Мікропроцесорні та програмні
засоби автоматизації”

Керування кроковим двигуном
за допомогою PIC16

Методичні вказівки до лабораторної роботи №8 “Керування кроковим двигуном за допомогою PIC16” з курсу “Мікропроцесорні та програмні засоби автоматизації” Медвідь В.Р., Пісьціо В.П., - Тернопіль: ТНТУ, 2018 - 17 с.

Відповідальні за випуск

доцент, к.т.н. Медвідь В.Р.,

асистент Пісьціо В.П.

Для студентів напрямку: 151 "Автоматизація та комп'ютерно-інтегровані технології"

Лабораторна робота №8

Керування кроковим двигуном за допомогою PIC16

1. Послідовність роботи з програмним симулятором PIC Simulator IDE

Основне вікно програми PIC Simulator IDE має вигляд, показаний на (рис. 1).

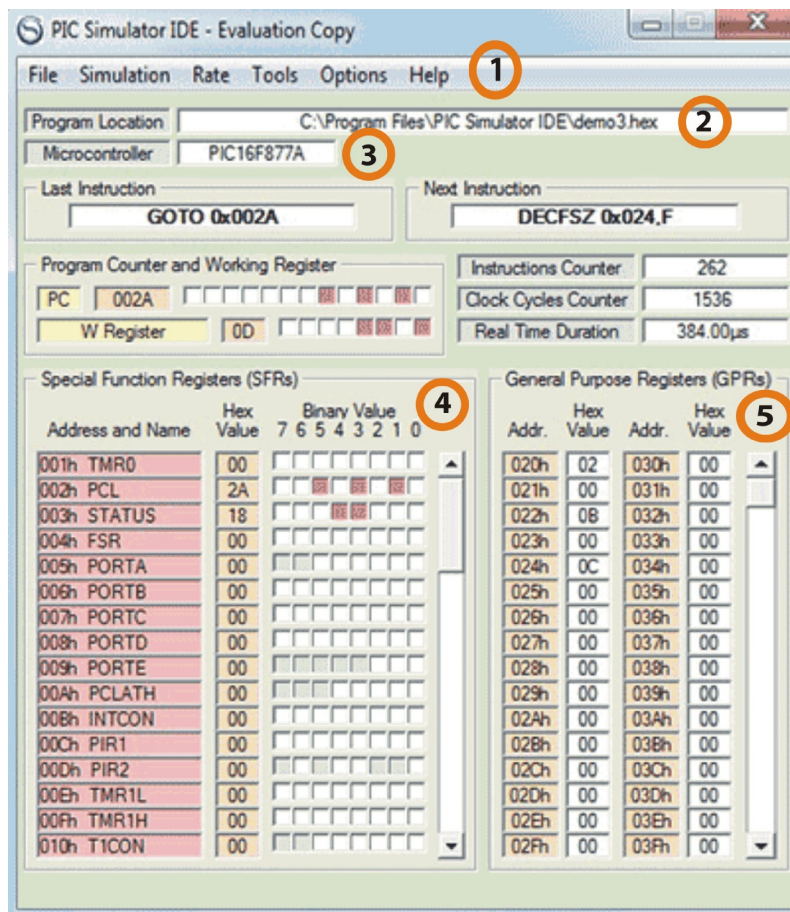


Рис. 1. Основне вікно програми PIC Simulator IDE

У верхній частині знаходяться меню, через які можна отримати доступ до основних і додаткових модулів програми (поз. 1)(рис. 1).

В рядку Program Location вказано шлях до обраної програми і її ім'я (поз. 2).

В рядку Microcontrollers, відображається тип обраного мікроконтролера (поз. 3).

У нижній частині вікна є дві панелі (поз.4 і поз.5), в яких відображаються стан програми, вміст регістрів спеціальних функцій (РСФ) і керуючих регістрів обраного МК.

Послідовність роботи з програмним симулятором наступний:

- запуск програми PIC Simulator IDE;
- вибір типу мікроконтролера, для якого написана програма;
- вибір частоти кварцового генератора (впливає тільки на відображувані програмою дані про час виконання програми або команди, але не на швидкість роботи програми, що налагоджуються в PIC Simulator IDE);
- завантаження програми у вигляді HEX-файлу або запуск вбудованого компілятора мови асемблера і написання в ньому потрібної програми;
- вибір потрібних модулів віртуальних пристроїв;
- вибір швидкості і режиму роботи програми симулятора;
- запуск процесу симуляції роботи програми на обраному МК.

Якщо потрібно скористатися для роботи з симулятором власною програмою або внести зміни у вже розроблену, необхідно створити або завантажити для цього файл асемблера, з якого після компіляції буде створений необхідний для роботи з симулятором hex-файл.

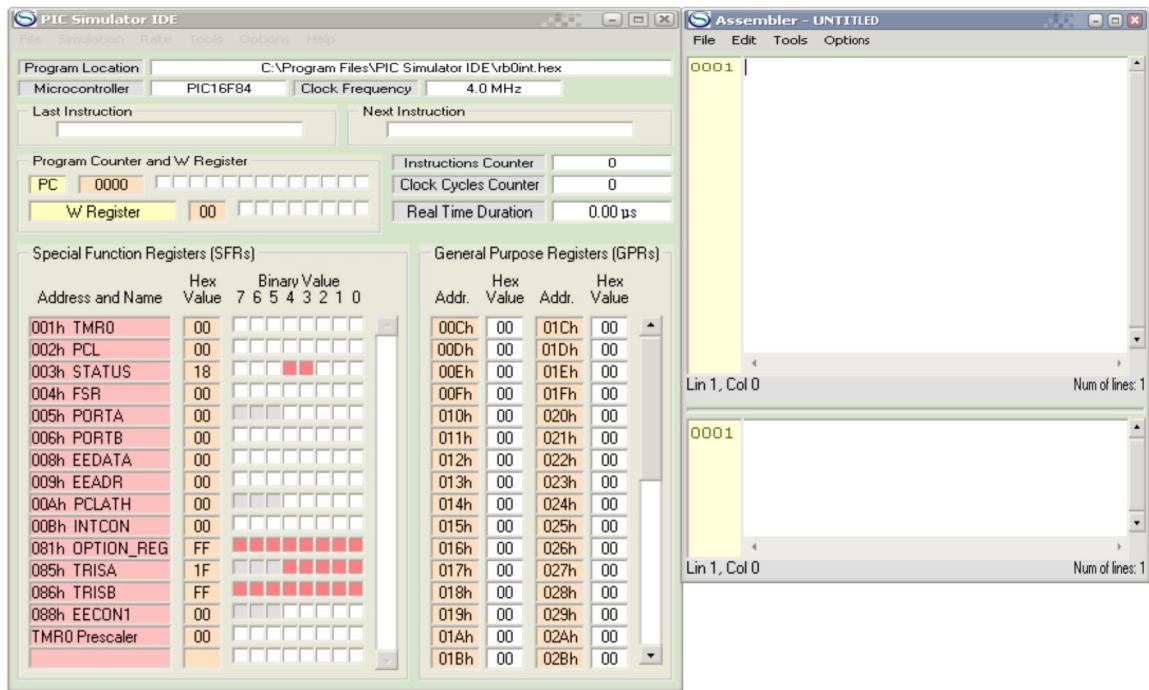


Рис. 2. Вікно симулятора з відкритим вікном Assembler

Для цього:

1. Натиснути Options | Assembler. Відкриється вікно компілятора Assembler – UNTITLED (рис. 2);
2. У вікні Assembler натиснути опцію File. Розкриється закладка (рис. 3), з якої для створення нового файлу потрібно натиснути New, а для завантаження вже створеного – OPEN.

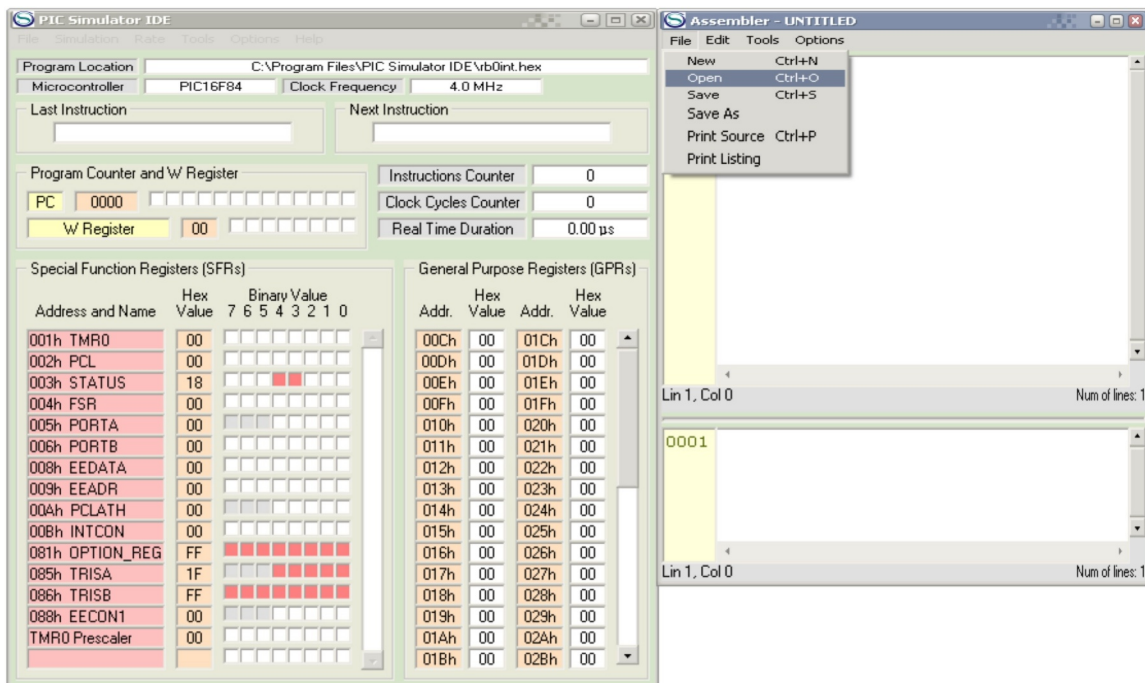


Рис. 3. Завантаження існуючого або створення нового файлу асемблера

3. Після вибору і завантаження файлу (наприклад, gb0int.asm), його текст з'явиться у вікні Assembler (рис. 4).

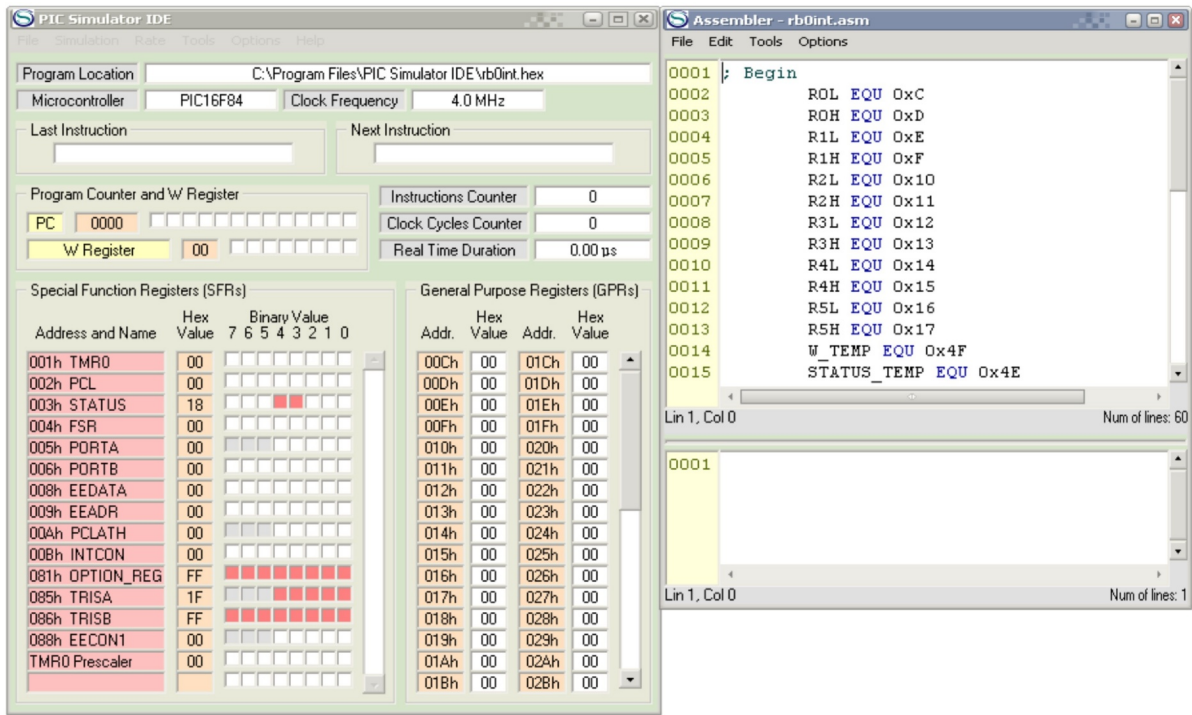


Рис. 4. Завантаження файлу rb0int.asm

4. Для компіляції створеного або завантаженого і потім зміненого файлу, натисніть Tools і у вікні, що розкриється – Assemble. В нижній половині вікна Assembler з'явиться лістинг відкомпільованого файлу і, одночасно, при відсутності помилок, буде створений одноіменний hex-файл.

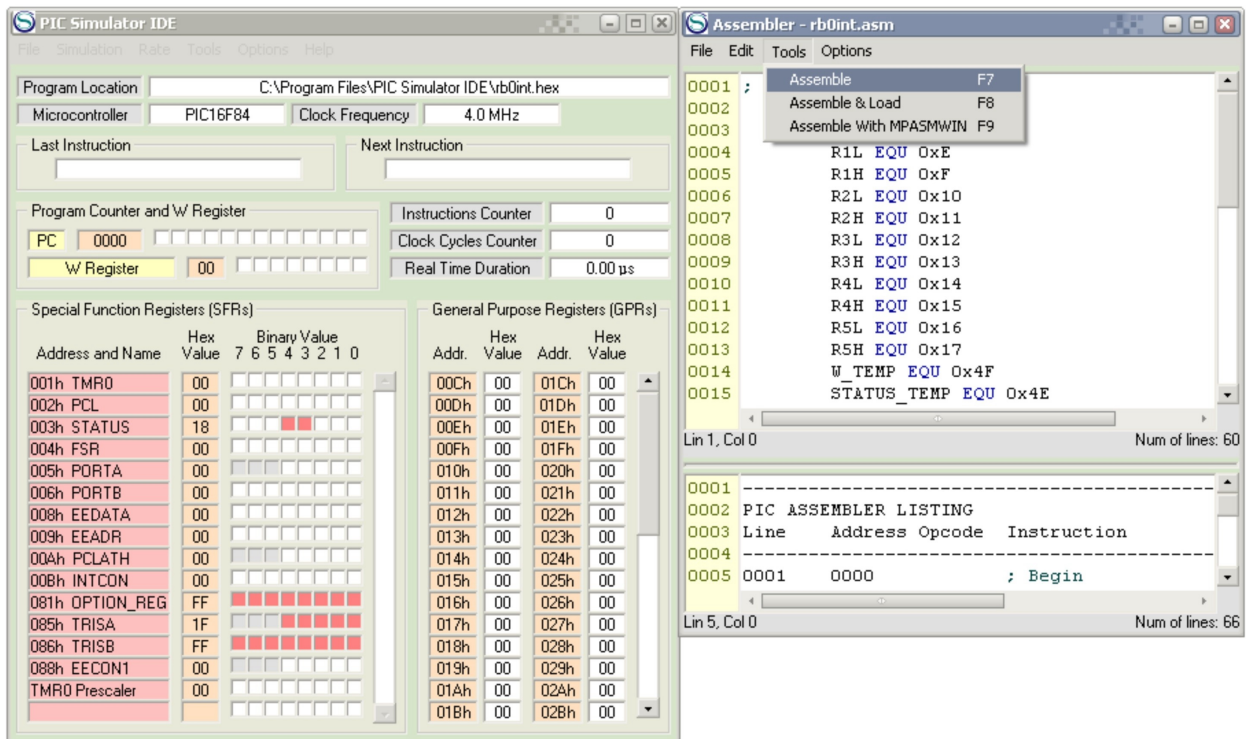


Рис. 5.

2. Режими керування біполярним кроковим двигуном

2.1. Типи крокових двигунів

Кроковий двигун (КД) являє собою безколекторний двигун постійного струму з фіксованими положеннями валу.

КД призначено для точного позиціювання валу без застосування систем зворотного зв'язку. Обмотки КД є частиною статора. На роторі розташовано постійний магніт або, у випадках зі змінним магнітним опором, зубчастий блок з магнітом якого матеріалу.

Усі комутації проводяться за зовнішніми схемами керування. **На двигунах з постійними магнітами звичайно є дві незалежні обмотки.**

Крокові двигуни мають широкий діапазон кутів дозволів. Більш грубі двигуни, звичайно с постійними магнітами, обертаються на 90° за крок, у той час як прецизійні двигуни можуть мати дозвіл $1,8^\circ$ або $0,72^\circ$ на крок.

«Обертове» магнітне поле забезпечується відповідними перемиканнями напруг на обмотках. Слідом за цим полем буде обертатися ротор, з'єднаний за допомогою редуктора з вихідним валом двигуна. Потужність крокових двигунів знаходиться у діапазоні від одиниць ватів до одного кіловату.

Кроковий двигун має не менш двох положень стійкої рівноваги ротора в межах одного оберту. Напряга живлення обмоток керування кроковим двигуном - це послідовність однополярних або двополярних прямокутних імпульсів, що надходять від електронного комутатора або контролера. Результуючий кут відповідає кількості перемикань комутатора, а частота обертання двигуна – частоті перемикань електронного комутатора.

На рис. 6 зображено положення ротора крокового двигуна залежно від комутації обмоток. Послідовно комутуючи **струм в обмотках відповідно до діаграм**, наведених на рис. 7, можна змусити обертатися вектор магнітного поля, а за ним і ротор, у прямій або зворотній послідовності. Від'ємне значення струму через обмотку на діаграмі відовідає логічному «1-0» та зворотньому «0-1» значенням напруги, прикладеної до обмотки. При такому керуванні двигун має 4 стійких стани на одному оберті ротора.

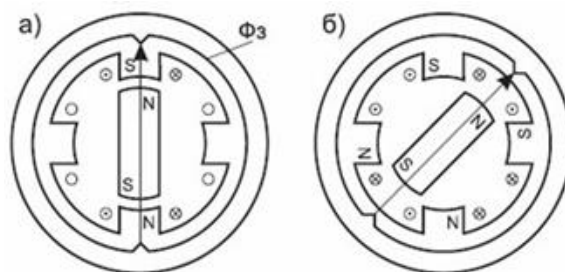


Рис. 6. Положення ротора при кроковому (а) і напівкроковому керуванні (б)

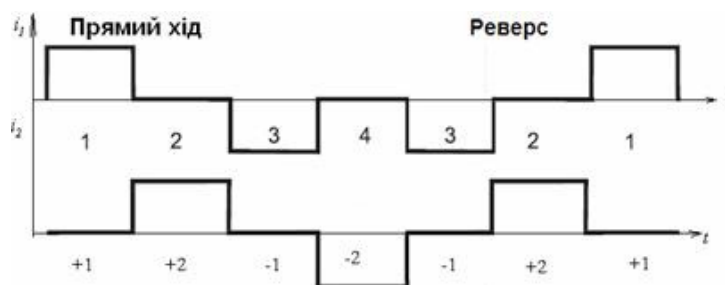


Рис. 7. Прямий і зворотний хід у кроковому двигуні

2.2. Біполярні і уніполярні крокові двигуни

В залежності від того, якою є форма обмоток крокового двигуна, двигуни діляться на **уніполярні і біполярні**.

У **біполярного** двигуна по 1 обмотці в кожній фазі, тобто всього дві обмотки і відповідно 4 виводи (рис. 8,а). Для забезпечення обертання валу на ці обмотки має подаватися напруга із змінною полярністю. Тому для біполярного двигуна необхідний півмостовий або мостовий драйвер, забезпечений двополярним живленням.

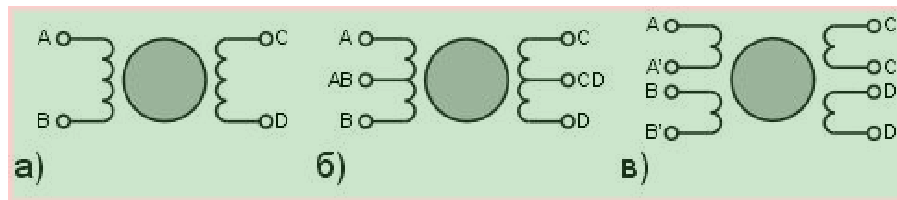


Рис. 8. Біполярні та уніполярні крокові двигуни

Уніполярний двигун також, як і біполярний, для кожної фази має по одній обмотці, але кожна обмотка містить відвід від середини. У зв'язку з цим, шляхом перемикання половинок обмоток крокового двигуна з'являється можливість міняти напрям магнітного поля (8,б).

Уніполярні двигуни можуть забезпечуватися чотирма обмотками, кожна з яких містить власні виводи - тобто їх всього вісім (рис. 8,в).

Уніполярний двигун, що має дві обмотки з відводами по середині, можна використовувати як біполярний. У цьому випадку проводи, що йдуть від середини обмоток, не використовуються.

Симулятор **Pic Simulator IDE** імітує роботу **уніполярного крокового двигуна**.

Він містить дві обмотки з середніми виводами, які заземляються (у вікні симулятора крокового двигуна заземлення середніх виводів не показано) (рис. 9).

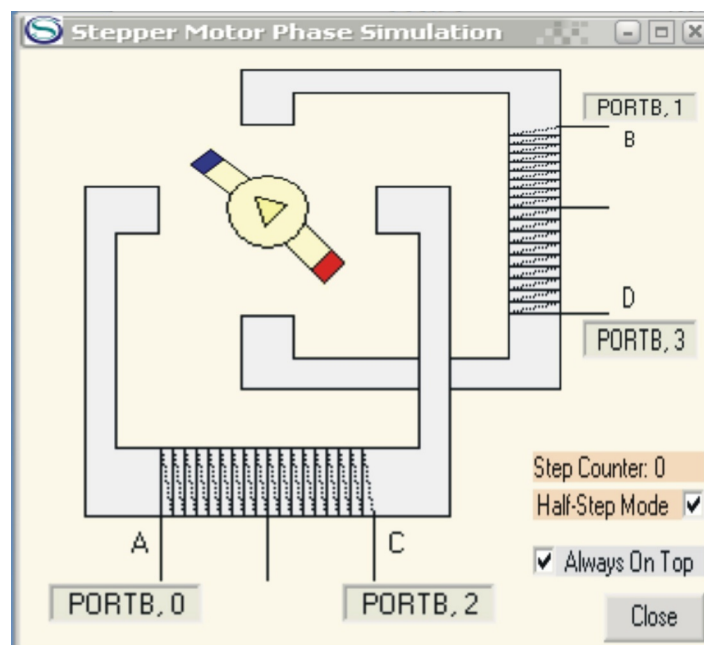


Рис. 9 Вікно «Stepper Motor Phase Simulation»

2.3. Основні способи управління кроковим двигуном

- **Озрізняють:**
 - повнокроковий режим без перекриття фаз,
 - повнокроковий режим з перекриттям фаз,
 - напівкроковий режим.

Повнокроковий режим без перекриття фаз

У цьому режимі (рис. 10) в один момент часу живиться тільки одна фаза двигуна. Полюси ротора займають положення навпроти обмотки, на яку подається живлення, залежно від напрямку протікання струму в ній. Недоліком даного способу є менший момент двигуна.

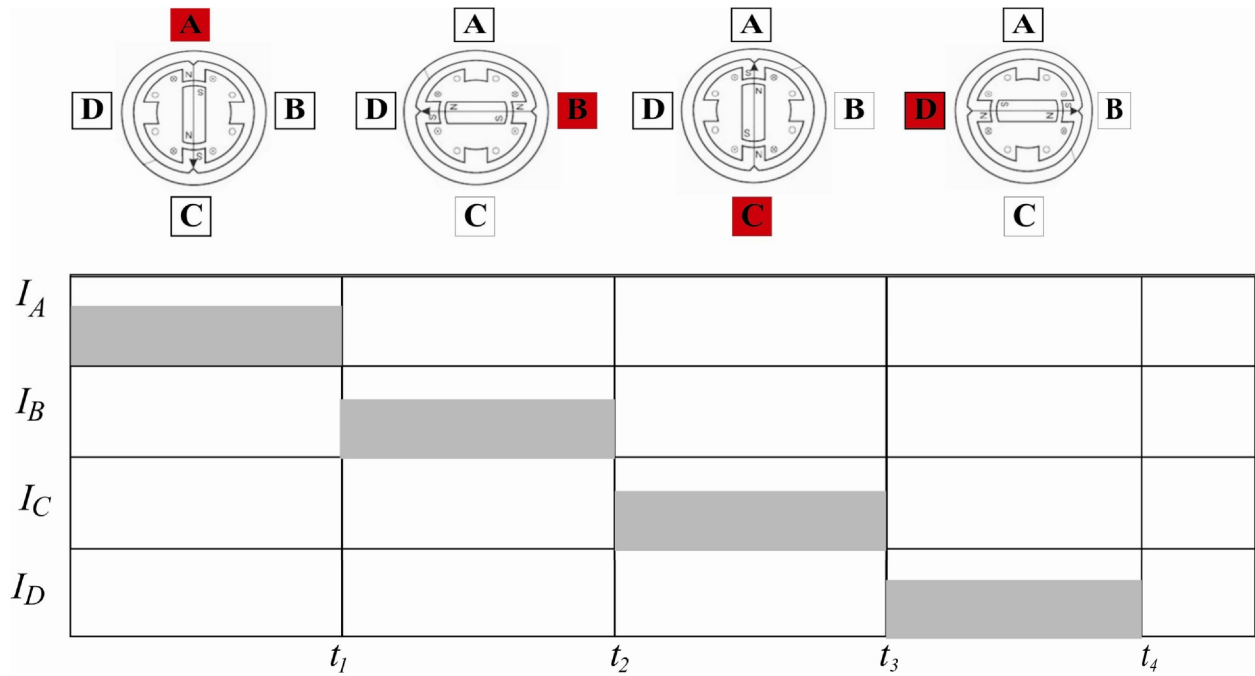


Рис. 10. Діаграми роботи в повнокроковому режимі без перекриття фаз для уніполярного двигуна

Повнокроковий режим з перекриттям фаз

Характерною рисою даного режиму є те, що одночасно подається живлення на дві суміжні фази, і ротор зупиняється не напроти полюсів, а в проміжному положенні між ними.

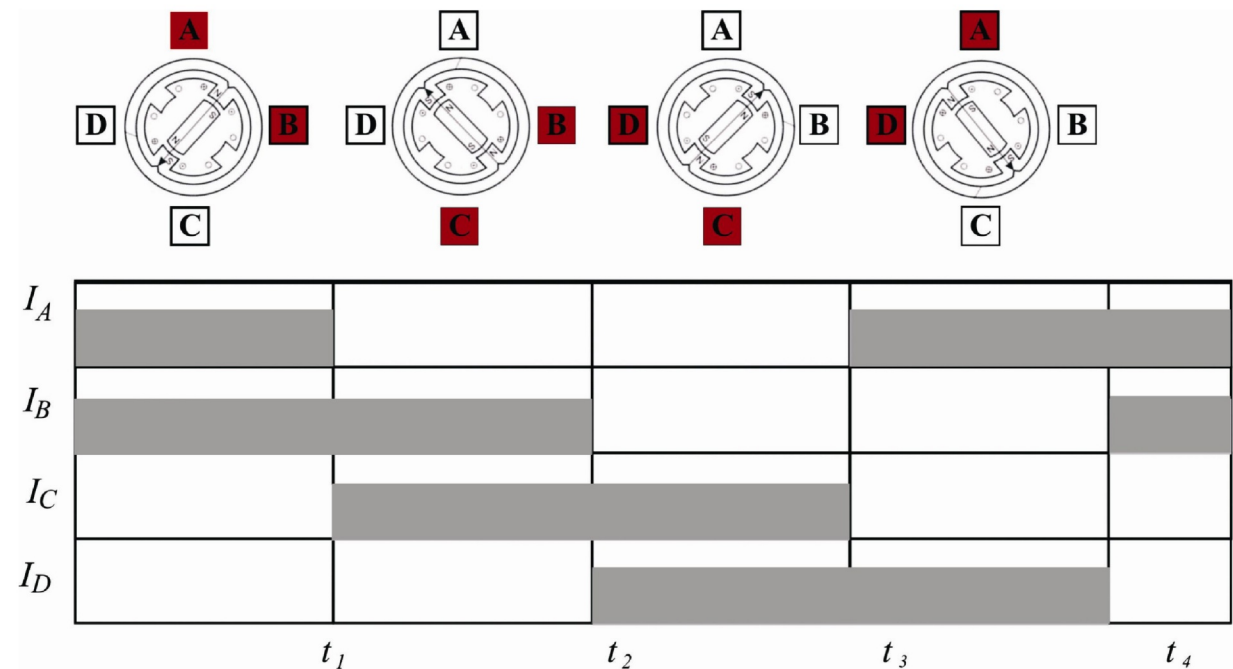


Рис. 11. Діаграми роботи в повнокроковому режимі з перекриттям фаз для уніполярного двигуна

Даний спосіб комутації фаз (рис. 11) забезпечує в $2^{1/2}$ разів більший момент. При зупинці двигуна важливо не знеструмлювати його обмотки, щоб двигун забезпечував повний момент, бо може статися зсув ротора на половину кроку і відповідно втрата положення.

Напівкроковий режим

Є комбінацією двох вищеназваних, тобто ротор зупиняється як навпроти полюсів, так і в проміжному положенні між ними. З одного боку, це дозволяє зменшити крок у два рази, з іншого - будуть коливання моменту, тому що коли на дві фази подається живлення, то момент буде в $2^{1/2}$ разів більшим.

Щоб уникнути цього явища, в момент, коли включені дві фази, необхідно занижувати на них струм теж у $2^{1/2}$ разів більший, але це призведе до загального зменшення моменту.

Проблемою напівкрокового режиму є перехід в стан з однією включеною фазою. В цьому випадку потрібно якомога швидше звести в фазі, яка вимикається, струм до нуля.

Коли використовується мостова схема, це здійснюється вимиканням всіх ключів, що призводить до того, що великий розрядний струм протікає через діоди і джерело живлення.

Якщо ж залишити один ключ замкненим, то коло розряду буде включати в себе діод і ключ, тому струм буде спадати повільніше.

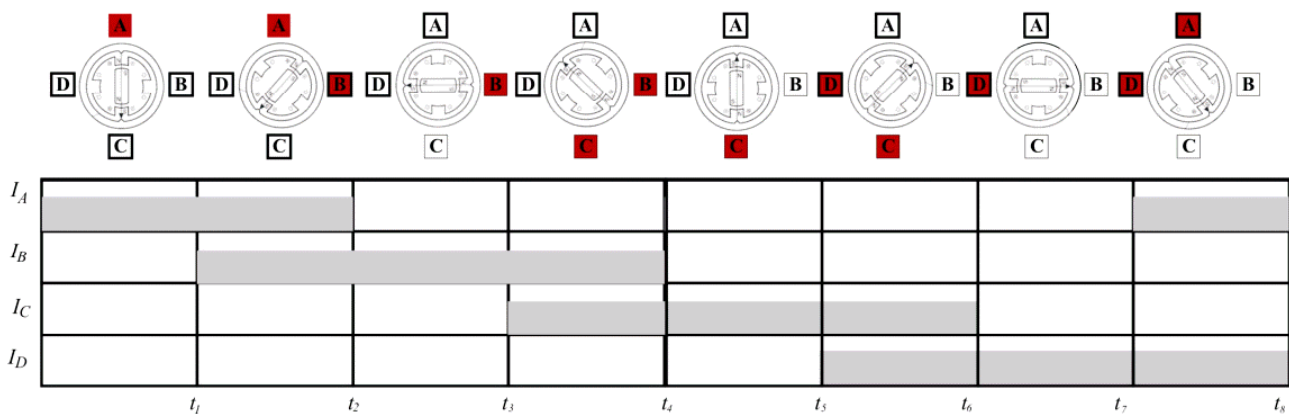


Рис. 12. Діаграми роботи в напівкроковому режимі для уніполярного двигуна

Діаграму, зображену на рис. 10 у вигляді **покрокової залежності струмів через обмотки**, зобразимо у вигляді логічних рівнів напруг на виводах порту В на кожному з кроків роботи двигуна протягом двох обертів ротора (табл. 1).

Таблиця 1

	PORTB.3	PORTB.2	PORTB.1	PORTB.0
Крок 1	«0»	«0»	«0»	«1»
Крок 2	«0»	«0»	«1»	«0»
Крок 3	«0»	«1»	«0»	«0»
Крок 4	«1»	«0»	«0»	«0»
Крок 5	«0»	«0»	«0»	«1»
Крок 6	«0»	«0»	«1»	«0»
Крок 7	«0»	«1»	«0»	«0»
Крок 8	«1»	«0»	«0»	«0»
.....

Дослідимо роботу такого двигуна в повнокроковому режимі без перекриття фаз.

Приклад 1

Програма забезпечує керування кроковим двигуном, який під'єднаний до чотирьох молодших розрядів порту В (рис. 9).

Текст програми з робочого файлу має наступний вигляд:

```
; Begin
  ORG 0x0000
  R0L EQU 0x20
  R0H EQU 0x21
  BCF PCLATH,3
  BCF PCLATH,4
  BSF STATUS,RP0
  MOVLW 0x80
  MOVWF OPTION_REG
  MOVLW 0x00
  MOVWF TRISB
  BCF STATUS,RP0
LB1:  MOVLW 0x03
      MOVWF R0L
      MOVLW 0x01
      MOVWF PORTB

LB2:  RLF PORTB,W
      MOVWF PORTB
      DECFSZ R0L,F
      GOTO LB2
      GOTO LB1
END
```

Послідовність роботи з симулятором при виконанні програми

Виконаємо цю програму в PIC Simulator ID, для чого необхідно:

1. Запустити PIC Simulator IDE;
2. Натиснути Options | Select Microcontroller;
3. Вибрати PIC16F84 і натиснути кнопку Select;
4. Натиснути Tools і у вікні, що розкриється, вибрати «Assembler». Відкриється вікно компілятора «Assembler – UNTITLED» (рис. 2);
5. Набрати текст програми Прикладу 1 у вікні «Assembler»;
6. Натиснути Tools і у вікні, що розкриється – Assemble. В нижній половині вікна Assembler з'явиться лістинг відкомпільованого файлу (рис. 13);
7. Одночасно, при відсутності помилок, буде створений файл з розширенням «.hex», для якого можна вибрати ім'я та шлях для запису. Записати його на «Робочий стіл» комп'ютера;
8. Вибрати File | Load Program і завантажити створений файл з розширенням «.hex»;
9. Натиснути Tools | Stepper Motor Phase Simulation. Відкриється вікно з панеллю крокового двигуна «Stepper Motor Phase Simulation» (рис. 14);
10. Вибрати Rate | Normal;

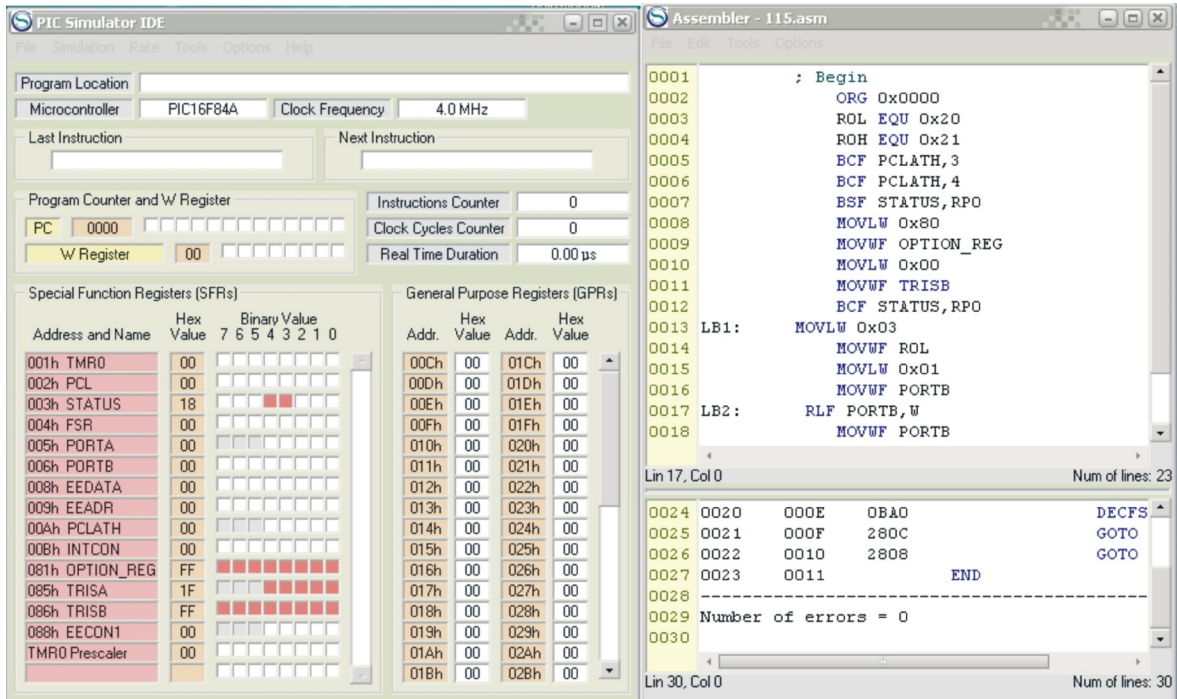


Рис. 13. Інтерфейс симулятора з виконуваною програмою та її лістингом після компілювання

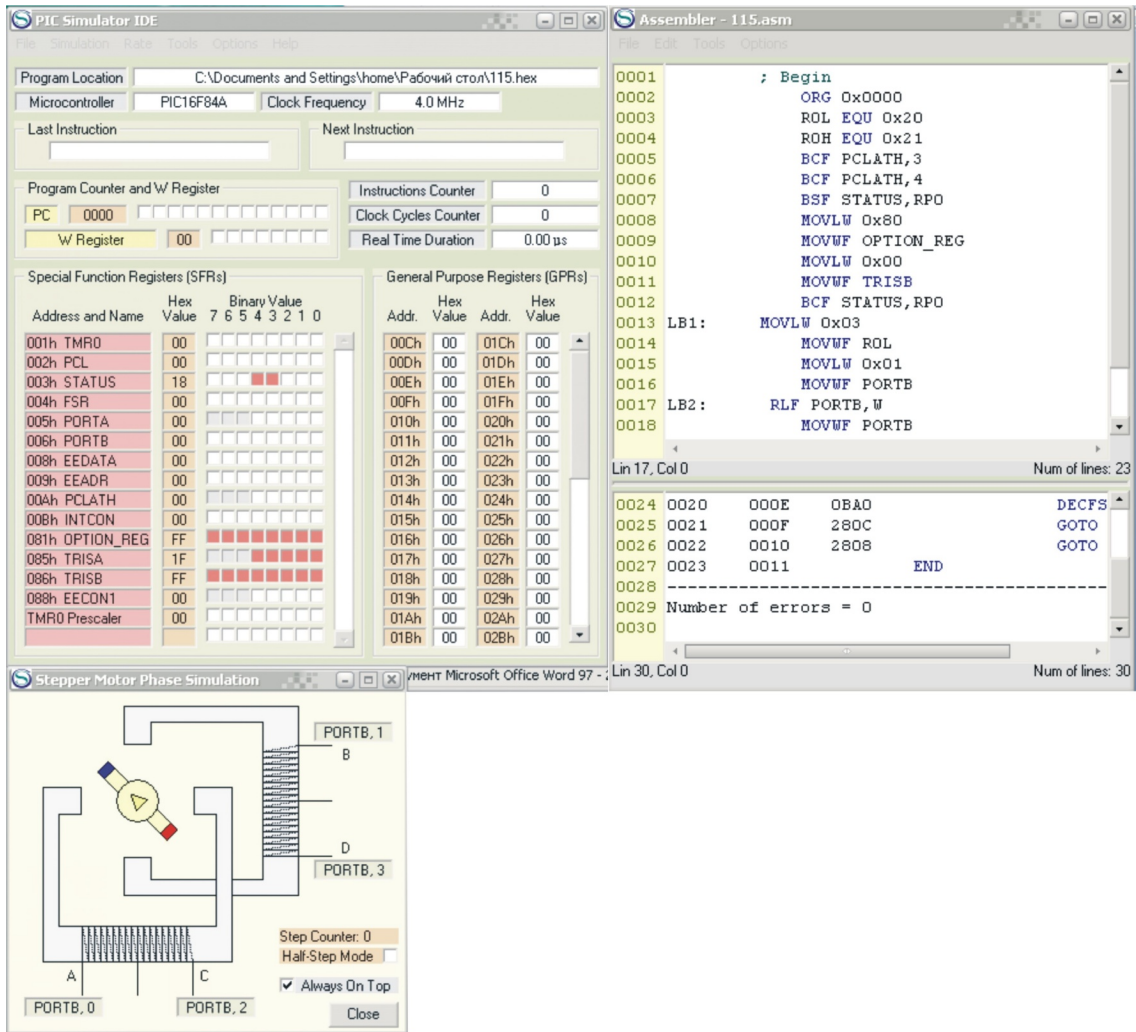


Рис. 14. Інтерфейс симулятора з виконуваною програмою та панеллю крокового двигуна «Stepper Motor Phase Simulation»

11. У вікні крокового двигуна почергово натиснути поле «PORTB,2» (вивід «С» крокового двигуна) і далі у вікні, що відкриється, вибрати вивід «1», після чого натиснути на поле «Select», яке розташоване внизу вікна. Таким чином, буде вибрано вивід «1» порту В.

Це ж повторити для виводу двигуна «В» («PORTB,1»), змінивши номер лінії порту на «PORTB,2»;

12. Натиснути Simulation | Start (почнеться виконання програми). Ротор двигуна почне обертатися з кроком 90^0 (двигун має 4 стійких стани на одному оберті ротора);

13. Щоб зупинити виконання програми, потрібно натиснути Simulation | Stop.

Для того, щоб мати змогу контролювати вміст регістрів після виконання симулятором кожної команди, перейти на виконання програми в кроковому режимі роботи.

Для цього:

1. В основному вікні симулятора натиснути Rate | Step By Step, а далі вибрати опцію Simulation і натиснути Start. Симулятор готовий до виконання програми в кроковому режимі;

2. Для виконання наступної команди програми потрібно натиснути на закладку STEP, яка з'явиться справа від закладки HELP вгорі основного вікна симулятора після вибору крокового режиму роботи програми.

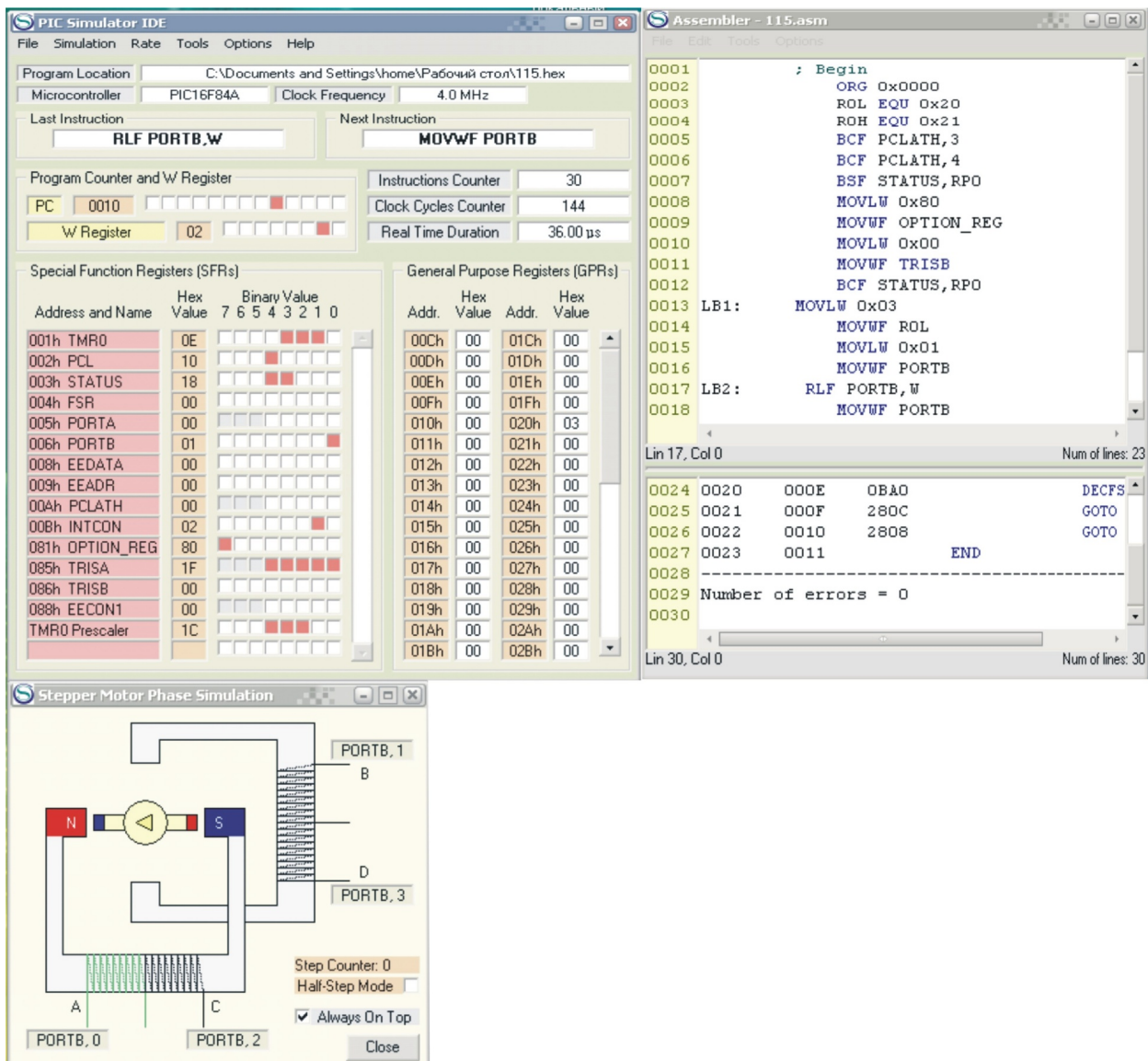


Рис. 15. Інтерфейс симулятора при виконанні програми керування кроковим двигуном «Stepper Motor Phase Simulation»

Вміст регістрів контролера, які використовуються при виконанні команд програми, знайти в області регістрів Address and Name, яка розташована в лівій нижній частині основного вікна симулятора (виділені рожевим кольором). Всі регістри восьмирозрядні.

В процесі виконання програми по зміні кольору комірок видно, вміст яких регістрів змінюється. Забарвлення комірки відповідного розряду регістру помаранчевим кольором означає наявність "1", білим - "0".

3. Завдання для виконання лабораторної роботи

Завдання 1

1. Виконати програму (Приклад1) в режимі «Normal». Переконайтеся в реалізації обертання ротора крокового двигуна в процесі виконання програми.

2. Виконати програму в повнокроковому режимі виконання програми. Вміст тих регістрів, значення яких змінюється в процесі виконання команд програми, записати в шістнадцятковому коді в табл. 2.

Таблиця 2

Регістр	PC	W	TMR0	STATUS	PCL	TRISA	TRISB	PCLATH	EEADR	EEDATA	FSR
Команда 1											
Команда 2											
.....											
Команда n											

3. З програм Прикладу 1 вибрати десять команд і за таблицею команд асемблера для PIC-контролера (таблиця 3) записати коментар щодо призначення цих команд (див. Приклад 2, де наведено такий запис для однієї команди).

Приклад 2

Код команди

118A

Команда

BCF PCLATH, 3

Виконувана операція (коментар)

; скинути в "0" 3-й біт регістра PCLATH

і т.д.

Завдання 2

1. Скласти програму, яка реалізує **реверсне обертання крокового двигуна в повнокроковому режимі роботи двигуна**, що ілюструється діаграмою на рис. 10.

2. Скласти таблицю за прикладом табл. 1, яка ілюструє зміну рівнів напруги на виводах порту В в процесі керування двигуном, за прикладом діаграми зміни струмів через обмотки двигуна (рис. 10).

3. Виконати програму в режимі «Normal». Переконайтеся в реалізації реверсного обертання ротора крокового двигуна в процесі виконання програми.

4. Виконати програму в кроковому режимі виконання програми.

5. Вміст тих регістрів, значення яких змінюється в процесі виконання команд програми, записати в шістнадцятковому коді в табл. 2.

6. З виконуваної програми вибрати десять команд і за таблицею команд асемблера для PIC-контролера (таблиця 3) записати коментар щодо призначення цих команд (див. Приклад 2, де наведено такий запис для однієї команди).

Завдання 3

1. Скласти програму, яка реалізує обертання крокового двигуна **в напівкроковому режимі** його роботи, що ілюструється діаграмою на рис. 12.

2. Скласти таблицю за прикладом табл. 1, яка ілюструє зміну рівнів напруги на виводах порту В в процесі керування двигуном, користуючись діаграмою зміни струмів через обмотки двигуна.

3. Виконати програму в режимі «Normal». Переконайтеся в реалізації обертання ротора

крокового двигуна з кроком в 45^0 в процесі виконання програми (двигун має 8 стійких станів на одному оберті ротора).

4. Виконати програму в кроковому режимі виконання програми.

5. Вміст тих регістрів, значення яких змінюється в процесі виконання команд програми, записати в шістнадцятковому коді в табл. 2.

6. З виконуваної програми вибрати десять команд і за таблицею команд асемблера для PIC-контролера (таблиця 3) записати коментар щодо призначення цих команд (див. Приклад 2, де наведено такий запис для однієї команди).

4. Контрольні запитання

1. Будова PIC-контролерів серії PIC16X8X.

2. Призначення регістрів мікроконтролера.

3. Формат та призначення регістрів PC, W, TRISHA, TRISHB, PORTA, PORTB.

4. Як програмуються лінії портів на ввід та на вивід?

5. Як організувати циклічний зсув розряду на виводах порту мікроконтролера?

6. Які режими роботи крокового двигуна можуть бути реалізовані при його керуванні?

Пояснити, в чому відмінність цих режимів.

7. Призначення та позначення основних елементів програмної моделі мікроконтролера.

5. Література

1. Данилин А. Програма-симулятор PIC Simulator IDE / Данилин А. // Современная электроника. 2006.- №4. -С. 68-76.

2. Тавернье К. PIC-микроконтроллеры. Практика применения. М.: ДМК, 2002.

3. Предко М. Создайте работа своими руками на PIC- контроллере./ Майкл Предко; Пер. с английского Земского Ю.В. – М.: ДМК Пресс, 2006. – 408 с.: ил.

4. Кениг А. и М. Полное руководство по PIC-микроконтроллерам.: Пер. с нем.-К.: МК-Пресс”, 2007.-256 с., ил.

Таблиця 3. Команди контролера PIC16X

Позначення	Функція	Цикли	Код команди	Біти стану	Прим.
ADDLW	Додавання константи і W	1	11111x	C, DC, Z	
ADDWF	Додавання W с f	1	00 0111 dfff ffff	C, DC, Z	1, 2
ANDLW	Логічне І константи і W	1	11 1001 kkkk kkkk	Z	
ANDWF	Логічне І W і f	1	00 0101 dfff ffff	Z	1, 2
BCF	Скидання біту в регістрі f	1	01 00bb bfff ffff		1, 2
BSF	Встановлення біту в регістрі f	1	01 01bb bfff ffff		1, 2
BTFSC	Пропустити команду, якщо біт у f дорівнює нулю	1 (2)	01 10bb bfff ffff		3
BTFSS	Пропустити команду, якщо біт у f дорівнює одиниці	1 (2)	01 11bb bfff ffff		3
CALL	Виклик підпрограми	2	10 0kkk kkkk kkkk		
CLRF	Скидання регістру f	1	00 0001 1fff ffff	Z	2
CLRW	Скидання регістра W	1	00 0001 0xxx xxxx	Z	
CLRWDT	Скидання сторожового таймера WDT	1	00 0000 0110 0100		
COMF	Інверсія регістру f	1	00 1001 dfff ffff	Z	1, 2
DECF	Декремент регістру f	1	00 0011 dfff ffff	Z	1, 2
DECFSZ	Декремент f, пропустити команду, якщо 0	1 (2)	00 1011 dfff ffff		1, 2, 3
GOTO	Перехід за адресою	2	10 1kkk kkkk kkkk		
INCF	Інкремент регістру f	1	00 1010 dfff ffff	Z	1, 2
INCFSZ	Інкремент f, пропустити команду, якщо 0	1 (2)	00 1111 dfff ffff		1, 2, 3
IORLW	Логічне АБО константи і W	1	11 1000 kkkk kkkk	Z	
IORWF	Логічне АБО W і f	1	00 0100 dfff ffff	Z	1, 2
MOVF	Пересилання регістру f	1	00 1000 dfff ffff	Z	1, 2
MOVLW	Пересилання константи в W	1	11 00xx kkkk kkkk		
MOVWF	Пересилання W у f	1	00 0000 1fff ffff		
NOP	Пуста команда	1	00 0000		

OPTION	Завантаження регістру OPTION	1	00 0000 0110 0010		
RETFIE	Повернення з переривання	2	00 0000 0000 1001		
RETLW	Повернення з підпрограми з заванта- женням константи в W	2	11 01xx kkkk kkkk		
RETURN	Повернення з підпрограми	2	00 0000 0000 1000		
RLF	Зсув f вліво через перенесення	1	00 1101 dfff ffff	C	1, 2
RRF	Зсув f вправо через перенесення	1	00 1100 dfff ffff	C	1, 2
SLEEP	Перехід у режим SLEEP	1	00 0000 0110 0011		
SUBLW	Вирахування W з константи	1	11 110x kkkk kkkk	C, DC, Z	
SUBWF	Вирахування W з f	1	00 0010 dfff ffff	C, DC, Z	1, 2
SWAPF	Обмін місцями тетрад в f	1	00 1110 dfff ffff		1, 2
TRIS	Завантаження регістру TRIS	1	00 0000 0110 0fff		
XORLW	Виключаюче АБО константи і W	1	11 1010 kkkk kkkk	Z	
XORWF	Виключаюче АБО W і f	1	00 0110 dfff ffff		1, 2

*Якщо в результаті виконання команди змінюється лічильник команд, або виконується перехід по перевірці умови, то команда виконується за два цикли. Другийцикл виконується як NOP.

Програма, що реалізує **напівкроковий** режим обертання **уніполярного крокового двигуна**:

```
;Begin
    ORG 0x0000
    R0L EQU 0x20
    R0H EQU 0x21
    BCF PCLATH,3
    BCF PCLATH,4
    BSF STATUS,RP0
    MOVLW 0x80
    MOVWF OPTION_REG
    MOVLW 0x00
    MOVWF TRISB
    BCF STATUS,RP0
    MOVLW 0x20
    MOVWF R0L
LB1:  MOVLW 0x01
    MOVWF PORTB
    MOVLW 0x03
    MOVWF PORTB
    MOVLW 0x02
    MOVWF PORTB
    MOVLW 0x05
    MOVWF PORTB
    MOVLW 0x04
    MOVWF PORTB
    MOVLW 0x0C
    MOVWF PORTB
    MOVLW 0x08
    MOVWF PORTB
    MOVLW 0x09
    MOVWF PORTB

LB2:  DECFSZ R0L,F
    GOTO LB1
    GOTO LB2
END
```